Sveučilište u Rijeci – Odjel za informatiku

Diplomski studij informatike: Informacijski i komunikacijski sustavi

Ernest Marnika

Postupci određivanja semantičke sličnosti tekstova

Diplomski rad

Mentor: izv. prof. dr. sc. Ana Meštrović

Rijeka, lipanj 2019.

ODJEL ZA INFORMATIKU
SVEUČILIŠTE U RIJECI

Rijeka, 3.6.2019.

Zadatak za diplomski rad

Pristupnik: Ernest Marnika

Naziv diplomskog rada: Postupci određivanja semantčke sličnosti tekstova

Naziv diplomskog rada na eng. jeziku: Algorithms for determing semantic similarity of texts

Sadržaj zadatka:

U području računalne analize prirodnog jezika definirani su brojni postupci mjerenja semantičke sličnosti tekstova. Među tradicionalnim postupcima kao najbolji ističe se latentna semantička analiza. Najnoviji postupci baziraju se na primjenama modela dubokog učenja (word2vec, GloVe, ELMO, i dr.) i pokazali su se kao potencijalno bolji odabir za mjerenje semantičke sličnosti tekstova. Zadatak diplomskog rada je dati pregled pojmova i postupaka vezanih za određivanje semantičke sličnosti tektova. U okviru prkatičnog dijela potrebno je usporediti jedan tradicionalni pristup i jedan pristup temeljen na modelu dubokog učenja na odabranom korpusu tekstova.

Mentor:

Izv. prof. dr. sc. Ana Meštrović

Voditelj za diplomske radove:

Izv. prof. dr. sc. Ana Meštrović

Komentor:

Zadatak preuzet: datum 03.06.2019.

(potpis pristupnika)

Sažetak

U radu je opisan pojam semantike, semantičkih mjera i sličnosti i vezanih pojmova semantici tekstova. Mjerenje semantičke sličnosti tekstova podrazumijeva pronalaženje sličnosti riječi tj. semantičkih odnosa temeljem nekih od modela opisanih u radu. Opisani modeli koriste biblioteku programskog modula Gensim (Library, G.), programski jezik Python nadograđen do zadnje verzije, a dodatno neki od njih koriste i neuronske mreže. Korišteni modeli su Doc2vec, Word2vec, i LSA. Cilj je pokazati koji od njih daje najtočnije i najbolje rezultate semantičke sličnosti na korpusu i odrediti koji je najbolji za korištenje u mjerenju sličnosti. Korišteni korpusi u radu su nazvani test.txt, test2.txt i text8.

Prilozi dodani ovom radu su programski kodovi opisanih modela stavljeni uz odgovarajuću implementaciju modela u radu te se poglavlja implementacija modela LSA, Word2vec i Doc2vec odnose na njih.

Ključne riječi

Semantika, semantičke mjere, korpus, semantičke sličnosti, Word2vec, GloVe, Doc2vec, LSA, mjerenje sličnosti.

Sadržaj:

1. Uvod

Polovicom dvadesetog stoljeća cilj umjetne inteligencije bio je dati mogućnost modernim računalima i strojevima da rješavaju kompleksne probleme i izvršavaju razne vrste zadatka. Slijedom navedenog, započinju se organizirati znanstvene konferencije i istraživanja iz područja umjetne inteligencije (AI) sa željom da se omogući računalima i strojevima da razumiju, uče, planiraju, komuniciraju i upravljaju znanjem (McCarthy, 2006). Jedan od ciljeva AI je odrediti sličnosti između koncepata i pojmova načinom na koji ljudi mogu usporediti objekte i situacije. Na taj bi način mogli izazvati razumijevanje i pružiti AI strojevima sposobnost prikupljanja znanja i vještina. U tom kontekstu, razni doprinosi bili su fokusirani na dizajniranje i učenje semantičkih mjera za usporedbu semantičkih entiteta kao što su riječi, rečenice ili tekstovi. Cilj ovih mjera je pristupiti sličnosti ili povezanosti tih entiteta preko njihove semantike. Određivanje semantičke sličnosti zasniva se na analizi semantičkih tekstualnih korpusa ili ontologija iz kojih se mogu izvlačiti mjere (Dict, 2012).

Pojam semantičke mjere nije uokviren matematičkom definicijom mjere. Shvaća se kao bilo koji teoretski alat, matematička funkcija, algoritam ili pristup koji omogućuje usporedbu semantičkih entiteta prema semantičkim dokazima, čak i ako postoji velika raznolikost mjera za procjenu sličnosti ili udaljenosti između određenih matematičkih objekata. Ove se mjere koriste za procjenu stupnja semantičke sličnosti između semantičkih entiteta kroz numeričku vrijednost. Ukoliko postoji velika raznolikost mjera za procjenu sličnosti ili udaljenosti između određenih matematičkih objekata, struktura podataka ili tipova podataka, glavna posebnost semantičkih mjera u usporedbi s tradicionalnim sličnostima ili funkcijama udaljenosti temelji se na više aspekata. Npr., mjere koje se koriste za usporedbu dviju riječi prema njihovim sljedovima znakova ne mogu se smatrati semantičkim jer se uzimaju u obzir samo znakovi riječi i njihov poredak, a ne njihovo značenje. Prema takvim sintaktičkim mjerama, riječi vozilo i auto smatrale bi se udaljenima unatoč svojoj srodnoj semantici. Iz tog razloga semantičke mjere oslanjaju se na analizu dvaju širokih tipova semantičkih modela: korpusa tekstova i ontologija. Modeli se koriste za izdvajanje semantičkih dokaza koji se koriste za semantičko mjerenje kako bi se podržala usporedba nekih usporedivih jedinica poput jezika, koncepata ili instanci.

U zadnjih nekoliko godina trendovi u mjerenju semantičke sličnosti idu u smjeru primjene modela dubokog učenja (engl. *Deep learning*) kao što su Word2vec, Doc2vec, GloVe, FastText i (Campr). Word2vec model i njegova ekstenzija FastText,+ riječi ili fraze iz korpusa preslikava u vektor različitih dimenzija ili realne brojeve, dok Doc2vec model preslikava rečenice ili paragrafe u vektorsko polje. Navedeni modeli treniraju neuronsku mrežu, stvaraju vokabular i pomoću biblioteke *gensim*, koja sadrži Python module tih modela, mogu biti implementirani za programsko testiranje na raznovrsnim korpusima ili tekstovima i dati puno bolje rezultate od statističkih modela poput latentne semantičke analize (LSA).

U ovome radu opisane su semantičke mjere i modeli mjerenja semantičke sličnosti. Konkretno, opisani su modeli koji se temelje na treniranoj neuronskoj mreži (Word2vec, Doc2vec, GloVe) i model koji se temelji na latentnoj semantičkoj analizi (LSA). U eksperimentalnom dijelu rada, izračunate su sličnosti nekih riječi iz korpusa koritenjem Word2vec-a i uspoređeni su rezultati mjerenja semantičke sličnosti rečenica primjenom LSA metode i modela Doc2vec na korpusu test2.txt, koji se sastoji od osam pozitivnih i šest negativnih rečenica uzetih iz glavnog korpusa test.txt koji predstavlja Microsoftov korpus parafraza sastavljen od pozitivnih i negativnih rečenica. Pozitivne se odnose na one koje parafraze jedna od druge, a negativne koje nisu. Primjer Doc2vec-a je i predstavljen na googleovom korpusu text8, koji sadrži podatke iz wikipedije.

U idućem su poglavlju opisane semantičke mjere, gdje i za što se koriste, njihova povezanosti sa semantičkim sličnostima, modelima sličnosti i njihova klasifikacija, a u ostalim poglavljima opisani su i implementirani modeli za usporedbu semantičkih sličnosti riječi u zadanom korpusu (LSA, Word2vec, Doc2vec, GloVe) te naposljetku usporedba njihovih rezultata i zaključak.

2. Semantičke mjere

Semantičke mjere (Harispe, 2017) koriste se za rješavanje problema u širokom rasponu aplikacija i domena. One su važan element za projektiranje brojnih algoritama i postupaka u kojima je semantika važna. Semantičke mjere su matematički alati kojima se procjenjuje snaga semantičkog odnosa između jedinica jezika, pojmova ili instanci, pomoću numeričkog opisa dobivenog prema

usporedbi informacija koje podržavaju njihovo značenje. Posebno se razmatraju tri područja primjene mjera semantičke sličnosti: računalna analiza prirodnog jezika, inženjerstvo znanja i semantički web (Ramzan, 2016) te biomedicinska informatika i bioinformatika. Budući da se radi o transverzalnim područjima, dodatno se razmatraju i aplikacije koje se odnose na pronalaženje i grupiranje informacija.

2.1.1. Računalna analiza prirodnog jezika

Računalna analiza prirodnog jezika (engl. *Natural Language Processing*, NLP) dio je informatike koji se bavi interakcijama između prirodnog jezika ljudi i računala. Neke od domena i problemi kojim se NLP bavi su klasifikacija tekstova, ekstrakcija informacija, strojno učenje, prepoznavanje i sinteza govora, sintaktička analiza i generiranje jezika, pronalaženje informacija i automatsko traženje određenih mjesta u korpusima (engl. *Information Retrieval*). Lingvisti su, sasvim prirodno, među prvima proučavali semantičke mjere u cilju uspoređivanja jezičnih jedinica (npr. riječi, rečenica, stavaka, dokumenata). Procjena povezanosti riječi i pojma igra važnu ulogu u otkrivanju parafraziranja; npr. duplikata sadržaja i plagiranja u generiranju tekstova, u sažetim tekstovima, u identificiranju strukture diskursa i dizajniranju sustava. Efikasnost semantičkih mjera za rješavanje sintaktičkih i semantičkih nejasnoća također je pokazana u nekoliko navrata.

2.1.2. Inženjerstvo znanja i semantički web

Zajednice povezane s inženjerstvom znanja (Ramzan, 2016), semantičkim i povezanim podacima igraju važnu ulogu u definiranju metodologija i standarda za formalno izražavanje strojno razumljivih reprezentacija znanja. Oni opsežno proučavaju problematiku povezanu s izražavanjem strukturiranih i kontroliranih rječnika, kao i ontologije, tj. formalne i eksplicitne specifikacije zajedničke koncepcije koja određuje skup pojmova, njihovih odnosa i aksioma za modeliranje domene. Ti se modeli oslanjaju na strukturirane reprezentacije znanja u kojima su semantika pojmova (klasa) i odnosa (svojstva) strogo i formalno definirani na nesvojstven način. Ova saznanja dovela su do definiranja nekoliko jezika koji se danas mogu koristiti za izražavanje formalnih, kompjuterski čitljivih i obradivih oblika znanja. Stoga su takvi modeli zamjenski izbor za usporedbu pojmova i primjera domene koju modeliraju. Taksonomija, znanost koja razvstava

pojmove na temelju sličnosti i razlika, posebno je korisna za procjenu stupnja sličnosti između pojmova. U tom području, semantičke mjere mogu se koristiti kao dio procesa koji imaju za cilj integriranje heterogenih ontologija; koriste se za pronalaženje sličnih ili dupliciranih entiteta definiranih u različitim ontologijama. Semantičke mjere uspješno su primijenjene i na zadatke učenja pomoću tehnologija semantičkog weba (Ramzan, 2016).

2.1.3. Biomedicinska informatika i bioinformatika

U biomedicinskoj informatici i bioinformatici definiran je velik broj semantičkih mjera. U tim domenama, semantičke mjere se obično koriste za proučavanje različitih tipova slučajeva koji su semantički karakterizirani pomoću ontologija (geni, proteini, lijekovi, bolesti). Nekoliko istraživanja vezanih uz korištenje semantičkih mjera u biomedicinskoj domeni naglašavaju raznolikost njihovih primjena, npr. za dijagnozu, klasifikaciju bolesti, dizajn lijeka i analizu. GO je preferirani primjer kojim se ističe veliko prihvaćanje ontologija u biologiji, opsežno se koristi za konceptualno označavanje gena (proizvoda) na temelju eksperimentalnih zapažanja ili automatskih zaključaka. Gen je klasično označen skupom koncepata strukturiranih u GO. Formalno su karakterizirani geni u pogledu njihovih molekularnih funkcija, njihovog staničnog položaja i bioloških procesa u koje su uključeni. Zahvaljujući semantičkim mjerama, ove omogućeno je automatsko uspoređivanje gena. Geni se mogu dalje analizirati razmatranjem njihove reprezentacije u semantičkom prostoru koji izražava naše sadašnje razumijevanje pojedinih aspekata biologije. U takvim slučajevima, konceptualne napomene premošćuju jaz između globalnog znanja o biologiji koje je definirano u GO-u (npr. organizacija molekularnih funkcija) i razumijevanja specifičnih primjera (npr. specifična uloga gena na molekularnoj razini). Semantičke mjere omogućuju računalima da iskoriste to znanje kako bi analizirali gene i stoga otvorili zanimljive perspektive za zaključivanje novog znanja. Primjerice, razne su studije naglasile važnost semantičkih mjera za procjenu funkcionalne sličnosti gena.

2.1.4. Ostale aplikacije

Prikupljanje informacija (IR) koristi semantičke mjere za prevladavanje ograničenja tehnika koje

se temelje na jednostavnom leksikografskom podudaranju pojma, tj. jednostavni IR modeli smatraju da je dokument relevantan prema upitu samo ako se uvjeti navedeni u upitu pojavljuju u dokumentu. Semantičke mjere omogućuju da se značenje riječi uzme u obzir prelaskom na sintaktičko pretraživanje, stoga se mogu koristiti za poboljšanje klasičnih modela, npr. sinonimi se više neće smatrati potpuno različitim riječima. Na primjer, semantičke mjere uspješno su korištene u dizajniranju ontoloških sustava za pronalaženje informacija i za proširenje upita. Važan aspekt je da semantičke mjere koje se temelje na ontologijama omogućuju analizu i ispitivanje resursa koji nisu tekstualni i stoga ne ograničavaju IR tehnike u analizi teksta, npr. geni koji su označeni pojmovima mogu se upitati. Također, definirano je rješenje generičkih indeksiranja temeljenih na semantičkim mjerama.

Geoinformatika aktivno doprinosi proučavanju semantičkih mjera. U ovom području mjere su korištene za izračun sličnosti između lokacija prema semantičkim karakterizacijama njihovih geografskih obilježja.

2.2. Semantičke sličnosti i mjere

2.2.1. Ljudska spoznaja, sličnost i postojeći modeli

Ljudska sposobnost procjenjivanja sličnosti stvari, npr. objekata, odavno je proučavana kognitivnim znanostima i psihologijom. Ona je okarakterizirana kao središnji element ljudskog kognitivnog sustava i stoga se danas shvaća kao ključni pojam za simulaciju inteligencije (Rissland. 2006). Ključni element je pokretanje procesa učenja u kojem nam sposobnost prepoznavanja sličnih situacija pomaže izgraditi naše iskustvo, aktivirati mentalne tragove, donositi odluke, inovirati ih primjenom iskustva stečenog u rješavanju sličnih. Važnost sličnosti za kognitivne procese, a posebno za proces učenja je naglašena teorijama transfera koje govore da se od novih vještina očekuje lakše učenje ako su slične vještinama koje su već naučene. Sličnost se smatra središnjom komponentom pronalaženja memorije, kategorizacije, prepoznavanja uzoraka, rješavanja problema, rezoniranja, kao i društvene prosudbe za povezane reference. Kognitivni modeli sličnosti općenito imaju za cilj proučavanje načina na koji ljudi procjenjuju

sličnost dvaju mentalnih reprezentacija u skladu s nekom vrstom psihološkog prostora. Temelje se na pretpostavkama o mentalnoj reprezentaciji uspoređenih objekata iz kojih će se procijeniti sličnost. Procjena sličnosti ne smije se shvatiti kao pokušaj uspoređivanja realizacije objekata kroz procjenu njihovih svojstava već kao proces koji ima za cilj usporediti objekte onako kako ih shvaća agent koji procjenjuje sličnost, npr. osoba. Pojam sličnosti ima smisla samo ako uzmemo djelomičnu reprezentaciju na kojoj se temelji procjena sličnosti objekta. Suprotno stvarnim objektima, prikazi objekata ne sadrže se u konačnim svojstvima. Npr., mentalne reprezentacije stvari obuhvaćaju ograničeni broj dimenzija objekta. Sličnost se na taj način može procijeniti između mentalnih reprezentacija. I za semantičke mjere sličnosti to može biti slučaj. Ukoliko se radi o egzistencijalnom zahtjevu reprezentacija za usporedbu stvari ili objekata, istraživanja o sličnosti u kognitivnim znanostima usredotočuje se na definiranje modela mentalne reprezentacije objekata, kako bi se dodatno razmotrile mjere koje će se koristiti za usporedbu objekata na temelju njihovih reprezentacije. Središnja uloga kognitivnih znanosti u proučavanju sličnosti ovisi o dizajniranju kognitivnih modela i mentalnih reprezentacija i sličnosti. Ti modeli koriste se za proučavanje načina na koji ljudi pohranjuju svoje znanje i komuniciraju s njim kako bi usporedili reprezentacije objekata. Kognitivni znanstvenici tada testiraju ove modele prema našem razumijevanju ljudskog uvažavanja sličnosti. Procjene ljudskog uvažavanja sličnosti pomažu nam razlikovati ograničenja ili očekivanja od svojstava koja bi trebao imati točan model. Neka istraživanja su pokazala da je uvažavanje sličnosti ponekad asimetrično. Sličnost između osobe i njegova portreta obično se očekuje da bude niža od suprotne. Očekivanje asimetrične procjene sličnosti je nespojivo s matematičkim svojstvima udaljenosti, koja je simetrična po definiciji. Modeli koji se temelje na aksiomima udaljenosti tako su se činili neadekvatnima i morali su se revidirati ili koristiti s umjerenošću. Uvode se kognitivni modeli sličnosti za razumijevanje temelja nekih pristupa usvojenih za definiranje semantičkih mjera.

Kognitivni modeli (Tversky, 2004) sličnosti organiziraju se u četiri različita modela:

1. Prostorni modeli
2. Modeli svojstava
3. Strukturni modeli
4. Transformacijski modeli

Prostorni modeli (Shepard, 1987), koji se također nazivaju geometrijskim modelima, oslanjaju se na jednu od najupečatljivijih teorija sličnosti u kognitivnim znanostima. Oni se temelje na pojmu psihološke udaljenosti i objekte smatraju točkama u višedimenzionalnom metričkom prostoru. Prostorni modeli razmatraju sličnost kao funkciju udaljenosti između mentalnih reprezentacija uspoređenih objekata. Predmeti su predstavljeni u višedimenzionalnom prostoru, a njihova su mjesta određena njihovim dimenzijskim razlikama. Statistička tehnika u obliku višedimenzionalnog skaliranja (MDS) koristi se za izvođenje nekih potencijalnih prostornih reprezentacija objekata iz obližnjih podataka (sličnost između parova objekata). Na temelju tih prostornih reprezentacija objekata izveden je univerzalni zakon generalizacije koji pokazuje da različite vrste podražaja (npr. Morseovi signali, oblici, zvukovi) imaju isti zakonit odnos između udaljenosti (u podvučenom MDS) i uočavaju mjere sličnosti. Sličnost između dva podražaja definirana je kao eksponencijalno padajuća funkcija njihove udaljenosti. Pokazujući negativan eksponencijalni odnos između sličnosti i generalizacije, uspostavljen je prvi zvučni model mentalne reprezentacije na kojem će kognitivne znanosti svoje studije temeljiti na sličnosti. Sličnost se pretpostavlja da je obrnuta udaljenost koja razdvaja perceptivne prikaze uspoređenih. Sličnost definirana kao funkcija udaljenosti je implicitno ograničena aksiomatskim svojstvima udaljenosti.

Model značajki (Tversky, 2004),je model u kojem se ocjenjivanim objektima manipulira kroz skupove značajki. Značajka opisuje bilo koju osobinu, karakteristiku ili aspekt objekata koji su relevantni za zadatak koji se istražuje. Modeli značajki procjenjuju sličnost dvaju podražaja u skladu s funkcijom podudaranja. Temelji se da su uobičajene i različite značajke dovoljne za njihovu usporedbu.

Strukturni modeli (Markman, 1993) temelje se na pretpostavci da su objekti predstavljeni strukturiranim prikazima. Npr. modeli strukturalnog poravnanja su modeli strukture u kojima se sličnost procjenjuje uporabom podudarnih funkcija koje će procijeniti podudarnost elemenata koji se uspoređuju. Postupak procjene sličnosti uključuje strukturalno usklađivanje dvaju mentalnih prikaza kako bi se razlikovale podudarnosti. Što je veći broj korespondencija, to će objekti biti sličniji. U nekim slučajevima, sličnost se procjenjuje na ekvivalentan način kao i analogno preslikavanje, a sličnost uključuje preslikavanja između obilježja i odnosa.

Transformacijski modeli (Hahn, 2003) pretpostavljaju da je sličnost određena transformacijskom distancom između mentalnih reprezentacija. Sličnost je uokvirena u reprezentacijsku distorziju i procijenjena je na temelju analize modifikacija potrebnih za transformaciju jednog prikaza u drugo. Sličnost se smatra smanjenom funkcijom transformacijske složenosti.

2.2.2. Semantičke mjere i vokabular

Cilj semantičkih mjera je uhvatiti snagu semantičke interakcije između semantičkih elemenata npr. riječi ili pojmova na temelju njihovog značenja. Jesu li riječi auto i automobil više semantički povezane od riječi auto i planina? Među-ljudski dogovor o ocjenama semantičke sličnosti je visok. Uvažavanje sličnosti je podložno višestrukim čimbenicima. Stariji ljudi i tinejdžeri vjerojatno neće povezati isti rezultat semantičke sličnosti između dva koncepta telefon i računalo. Ali, većinu vremena može se postići konsenzus oko procjene snage semantičke veze između elemenata. Većina semantičkih mjera pokušava oponašati ljudsku sposobnost da procijene stupanj povezanosti stvari prema semantičkim dokazima. Semantičke mjere procjenjuju snagu semantičkih interakcija između semantičkih entiteta prema analizi semantičkih modela (tekstovi, ontologije). Nisu sve mjere usmjerene na oponašanje ljudskog uvažavanja sličnosti. Dizajneri semantičkih mjera imaju za cilj samo uspoređivanje elemenata prema informacijama koje su definirane u semantičkom modelu, bez obzira na to jesu li rezultati dobiveni mjerenjem u korelaciji s ljudskim uvažavanjem semantičke sličnosti tj. povezanosti. U dizajniranju semantičkih mjera koje se temelje na ontologijama specifičnim za domenu to može biti slučaj. U tim slučajevima, ontologija se može povezati s našim razumijevanjem svijeta, ili domene, a semantička mjera može se smatrati našom sposobnošću da iskoristimo ovo znanje da bismo usporedili stvari. Cilj je biti koherentan sa znanjem izraženim u razmatranom semantičkom modelu, bez obzira na koherentnost modeliranog znanja. Primjer, semantička mjera koja se temelji na ontologiji koju su izradili stručnjaci za životinje ne bi smatrala da su dva koncepta ljenivac i majmun slični, iako većina ljudi misli da su lenjivci majmuni. S obzirom da semantičke mjere teže usporediti stvari prema njihovom značenju zarobljenom iz semantičkih dokaza, teško je dalje definirati pojam semantičkih mjera bez definiranja pojmova značenja i semantike. Pojam semantika se može definirati kao značenje ili tumačenje svih leksičkih jedinica, lingvističkih izraza ili primjera koji

su semantički karakterizirani prema određenom kontekstu. Semantičke mjere koriste se u više domena. Mogu se koristiti za vođenje usporedbe riječi-na-riječ, koncept-na-koncept, tekst-u-tekst ili primjer-na-primjer. Semantička mjera se definira kao funkcija:

$$\sigma k: E_k \times E_k \rightarrow \boldsymbol{R}$$

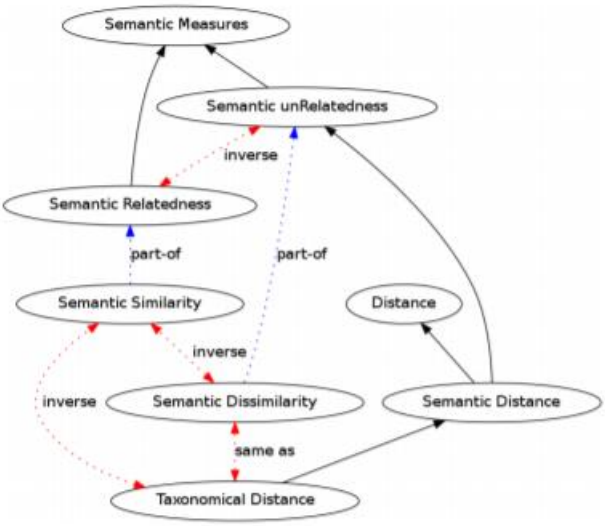
gdje je E_k skup elementa tipa k∈K, K su različiti tipovi elemenata koji se mogu usporediti s obzirom na njihovu semantiku, npr. K su: riječi, pojmovi, rečenice, tekstovi, web stranice, primjeri označeni pojmovima

Izraz se generalizira tako da se uzme usporedba različitih tipova elemenata. Važno je za procjenu uključivanja tekstova ili za usporedbu riječi i pojmova. Uspoređuju se parovi elemenata iste prirode. Kodomena funkcije σk može se ublažiti kako bi se razmotrile mjere koje proizvode rezultate definirane u složenije skale, npr. diskretne ili bipolarne skale. Radi jednostavnosti usredotočenost je na mjere koje su definirane u **R**. Semantičke mjere moraju implicitno ili eksplicitno iskoristiti semantičke dokaze. Npr. mjere usporedbe riječi kroz njihovu sintaktičku sličnost ne mogu se smatrati semantičkim mjerama jer se semantika odnosi na dokaze o značenju usporedivih elemenata. Razlika između pristupa koji se mogu i ne mogu asimilirati sa semantičkim mjerama ponekad je uska tj. ne postoji jasna granica razlikovanja nesemantike od semantičko-proširenih pristupa, već niz pristupa. Na primjer, netko može reći da mjere koje se koriste za procjenu sintaktičke udaljenosti između riječi obuhvaćaju semantičke dokaze vezane uz značenje riječi. Zapravo, niz znakova povezanih s jednom riječju proizlazi iz njegove etimologije koja se ponekad odnosi na njezino značenje, npr. riječi stvorene putem morfološke derivacije kao što je podskup iz skupa. Pojam semantičke mjere ponekad se teško razlikuje od mjera koje se koriste za usporedbu specifičnih struktura podataka. Neke semantičke mjere uspoređuju elemente koji su predstavljeni kroz kanonske oblike koji odgovaraju specifičnim strukturama podataka za koje su određene specifične nesemantičke mjere sličnosti. Na primjer, jedinice jezika predstavljene kao vektori mogu se uspoređivati pomoću mjera vektora sličnosti, ili se mogu koristiti čiste mjere sličnosti grafova za usporedbu entiteta definiranih u semantičkim grafovima. U nekim slučajevima, semantika mjere stoga nije obuhvaćena mjerom koja se koristi za usporedbu

kanonskih oblika elemenata. To je proces preslikavanja elementa iz semantičkog prostora (teksta, ontologije) u specifičnu podatkovnu strukturu (npr. vektor, skup), koja semantički poboljšava usporedbu. Teško je definirati definiciju rigorozne semantike pojma semantičke mjere. Tijekom godina semantičke mjere su se proučavale kroz različite pojmove, a ne uvijek u strogim uvjetima koristeći dobro definiranu terminologiju. Pojmovi koji se u literaturi obično upotrebljavaju za upućivanje na semantičke mjere su: semantička sličnost, semantička povezanost, semantička udaljenost, taksonomska udaljenost, semantička različitost, konceptualna udaljenost, itd. Ti pojmovi mogu imati različita značenja, ovisno o zajednicama ili autorima koji se odnose na njih. To naglašava poteškoće u definiranju i reduciranju tih pojmova u formalne matematičke okvire. Semantička povezanost je snaga semantičkih interakcija između dva elementa bez ograničenja na vrste razmatranih semantičkih veza. Pojam interakcije koji se koristi za definiranje semantičke povezanosti odnosi se na pozitivnu vrijednost, tj. što više dva elementa međusobno djeluju, više će se odnositi na njih. Na primjer u odnosu na semantičku povezanost, semantička udaljenost se odnosi na stupanj odbojnosti između dva uspoređena elementa. Semantička sličnost je podskup pojma semantičke povezanosti samo u razmatranju taksonomskih odnosa u procjeni semantičke interakcije između dva elementa. Semantičke mjere sličnosti uspoređuju elemente s obzirom na konstitutivna svojstva koja dijele i one koje su njima specifične. Dva koncepta čaj i šalica su stoga vrlo povezani, unatoč činjenici da nisu slični: pojam "Čaj" odnosi se na piće, a koncept šalica odnosi se na mjesto gdje se ulijeva čaj. Dva koncepta dijele nekoliko svojih konstitutivnih svojstava. To naglašava moguće tumačenje pojma sličnosti, što se može razumjeti u smislu supstitucije. Semantička sličnost od riječi do riječi ponekad se ne ocjenjuje samo na sinonime ili leksičke odnose koji se mogu smatrati ekvivalentnim taksonomskim vezama za riječi. Smatra se da procjena semantičke sličnosti dviju riječi mora uzeti u obzir i druge leksičke odnose, kao što antonimi[1]. U drugim slučajevima pojam semantičke sličnosti odnosi se na pristup koji se koristi za usporedbu elemenata, a ne na semantiku pridruženu rezultatima mjere. Npr., dizajneri semantičkih mjera koji se oslanjaju na ontologije koriste pojam semantičke sličnosti kako bi označili mjere koje se temelje na specifičnoj vrsti semantičke povezanosti npr. djelomični poredak koncepata definiranih odnosima između pojedinih dijelova. Semantika povezana s rezultatima povezanosti izračunata iz takvih ograničenja razlikuje se od semantičke sličnosti. Taksonomski odnosi koriste za procjenu semantičke sličnosti usporedivih elemenata. Stariji prilozi koji se

[1] Antonimi su prefiksalno suprotni leksemi riječi, npr. unijeti-iznijeti, otvoriti-zatvoriti.

odnose na semantičke mjere ne naglašavaju razliku između pojmova sličnosti i povezanosti. Na slici 1. je prikazana struktura različitih tipova semantika koji su povezani s semantičkim mjerama. Crni odnosi odnose se na taksonomske udaljenosti, obrnuti odnosi na semantičku interpretaciju pridruženu rezultatu mjere. Semantička sličnost i mjere različitosti imaju obrnute interpretacije.



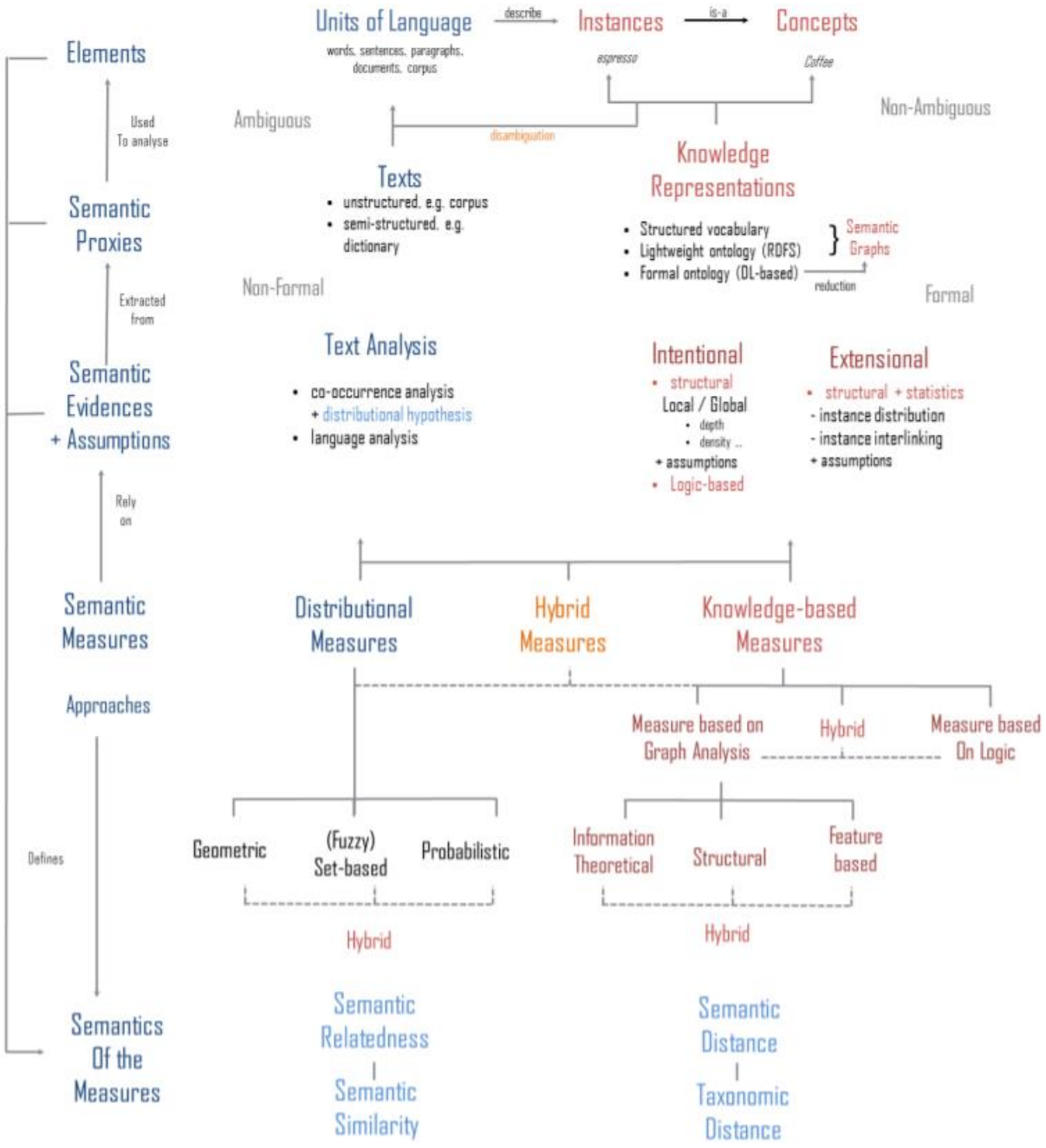
Slika 1. Neformalni semantički graf terminologije koja se odnosi na semantičke mjere (Harispe., 2017).

Ako gledamo udaljenost kao pojam sličnosti u semantičkim mjerama doprinosi se odnose na semantičke mjere se ne oslanjaju na formalne definicije pojma mjera ili udaljenosti. Doprinosi u vezi s semantičkim mjerama općenito se oslanjaju na uobičajeno prihvaćena očekivanja u vezi s tim pojmovima, tj. sličnost (tj. udaljenost) mora biti veća (odnosno niža). Pojmovi mjere i udaljenost u matematici su strogo definirani specifičnim aksiomima iz kojih proizlaze određena svojstva. Ti pojmovi izraženi su za dobro definirane objekte. Nekoliko se doprinosa oslanja na temelju tih aksiomatskih definicija. Opisuje matematičku pozadinu koja se odnosi na pojmove udaljenosti i sličnosti. To pripomaže da se strogo definira i bolje okarakteriziraju semantičke mjere u matematičkim terminima. Mjere udaljenosti i sličnosti formalno su definirane u matematici kao funkcije. Nije uvedena precizna i tehnička matematička definicija mjere predložena teorijom mjera. Pojam mjere koji se koristi nije uokviren strogim matematičkim definicijama mjere. Takva bi definicija isključila mnoge definirane semantičke mjere. Različite zajednice koristile su

koncepte sličnosti ili udaljenosti bez razmatranja aksiomatske definicije predložene u matematici, ali koristeći njihova široka intuitivna značenja. Semantičke mjere općenito se odnose na semantičku udaljenost kao na bilo koju ne-negativnu funkciju, osmišljenu da obuhvati suprotnu snagu semantičkih interakcija koje povezuju dva elementa. Što je veća snaga semantičkih interakcija između dva elementa, niža je udaljenost. Semantičku udaljenost definiramo kao funkciju koja procjenjuje semantičku nepovezanost. Pojam semantičke udaljenosti odnosi se na bilo koju funkciju osmišljenu da obuhvati semantičku nepovezanost. Funkcija poštuje ili ne poštuje aksiomatsku definiciju udaljenosti kada je to potrebno. Mjere semantičke povezanosti su funkcije koje su povezane s obrnutom semantikom jedne povezano s semantičkom nepovezanošću: što je veća snaga semantičkih interakcija između dva elementa, viša funkcija će procijeniti njihovu semantičku povezanost.

2.3. Klasifikacija semantičkih mjera

Na slici 2. djelomično je prikazan pregled modela klasifikacije semantičkih mjera koji se može koristiti za usporediti različite vrste semantičkih entiteta (npr. riječi, pojmove, instance). Sažima jednu od klasifikacije semantičkih mjera koje se mogu predložiti. Mjere prvo mogu biti klasificirane na temelju elemenata koje mogu usporediti. Na temelju tog aspekta razlikuju se dva glavna tipa mjera. Mjere koje se temelje na korpusu (engl. corpus-based) za usporedbu jedinica jezika, pojmova ili primjera iz analize teksta, tj. nestrukturirani semantički model. Te se mjere obično koriste za usporedbu riječi. One se također mogu prilagoditi za usporedbu pojmova ili primjera uzimajući u obzir da su tehnike razjašnjavanja korištene za identificiranje pojma ili primjera oznake u tekstovima. Druge su mjere koje se temelje na znanju (engl. knowledge-based) i koje su dizajnirane za uspoređivanje entiteta definiranih u ontologijama, tj. strukturirani semantički model. Mjere koje se temelje na znanju također se mogu koristiti za usporedbu jedinica jezika, npr. rečenica ili tekstova, na primjer ako uzmemo u obzir da tehnike razjašnjavanja imaju koristi se za uspostavljanje mostova između tekstova i ontologija.



Slika 2. Klasifikacija semantičkih mjera (Harispe, 2017)

Korpusne semantičke mjere omogućuju usporedbu jedinica jezika iz analize nestrukturiranih ili djelomično strukturiranih (engl. semi-structured) tekstova/podataka. Općenito se koriste za

usporedbu riječi, rečenica ili tekstova primjenom različitih NLP tehnika koje se najčešće oslanjaju samo na statističku analizu upotrebe riječi u tekstovima, npr. na temelju analize riječi pojavljivanja i jezičnog konteksta u kojem se pojavljuju. Mjere temeljene na korpusu ne mogu se u većini slučajeva svesti na jednu matematičku formulu. U tu svrhu, semantičke mjere temeljene na korpusu koriste prednosti velikog broja podataka informacija algoritma pronalaženja što čini semantičke mjere na temelju korpusa širokim područjem istraživanja na račvanju između nekoliko područja npr. računalne lingvistike i pronalaženje informacija. Mjere koje se temelje na korpusima često se označavaju kao mjere distribucije. Naglašava da je većina mjera eksplicitno ili implicitno utemeljena na raspodjeli hipoteza. Usvaja se općenitija oznaka semantičkih mjera temeljenih na korpusu. Cilj je olakšati uvođenje raznih mjera koje se temelje na korpusu ili analizi prirodnog jezika, uključujući one za koje se hipoteza distribucije ne smatra korijenom pristupa. Dokazano je da su sve korpusne semantičke mjere implicitno ili eksplicitno definirane u okviru distributivne semantike. Semantičke mjere bazirane na korpusu temelje se na strategiji na koja se koristiti za dohvaćanje značenje riječi. Značenje se često smatra funkcijom njegove uporabe u semantičkom prostoru izgrađenom iz korpusa tekstova. Ovisno o strategiji koja je usvojena za karakterizirati značenje riječi i predstavljanje semantičkog prostora u kojem je to značenje definirano, određeni kanonski oblik će biti odabran da predstavlja riječ. To omogućuje obradu značenja riječi algoritmima i može usporediti riječi na temelju usporedbe njihovih kanonskih oblika. Korpusne semantičke mjere su složeni procesi te postoje više algoritama za procjenu sličnosti. Premise koje definiraju značenje riječi, skup pretpostavki definiraju dokaz prirodnim jezikom iz kojeg se može preuzeti značenje riječi, prikazom riječi kao što je značenje ili algoritmom i funkcijom posebno definiranom za usporedbu prikaza dvaju riječi. Za razliku od korpusnih, semantičke mjere koje se temelje na znanju oslanjaju na više ili manje formalne izraze znanja definiranje načina na koji se moraju razumjeti usporedivi entiteti, tj. koncepti ili instance. Nisu ograničene za usporedbu jedinica jezika i mogu se koristiti za usporedbu bilo kojeg formalno opisanog dijela znanja, koji obuhvaća veliku raznolikost elemenata, npr. geni, osoba, glazbeni bendovi itd. Te se mjere najčešće koriste za usporedbu pojmova strukturiranih kroz semantičke odnose ili definirane pojmove u sustavima organizacije znanja.

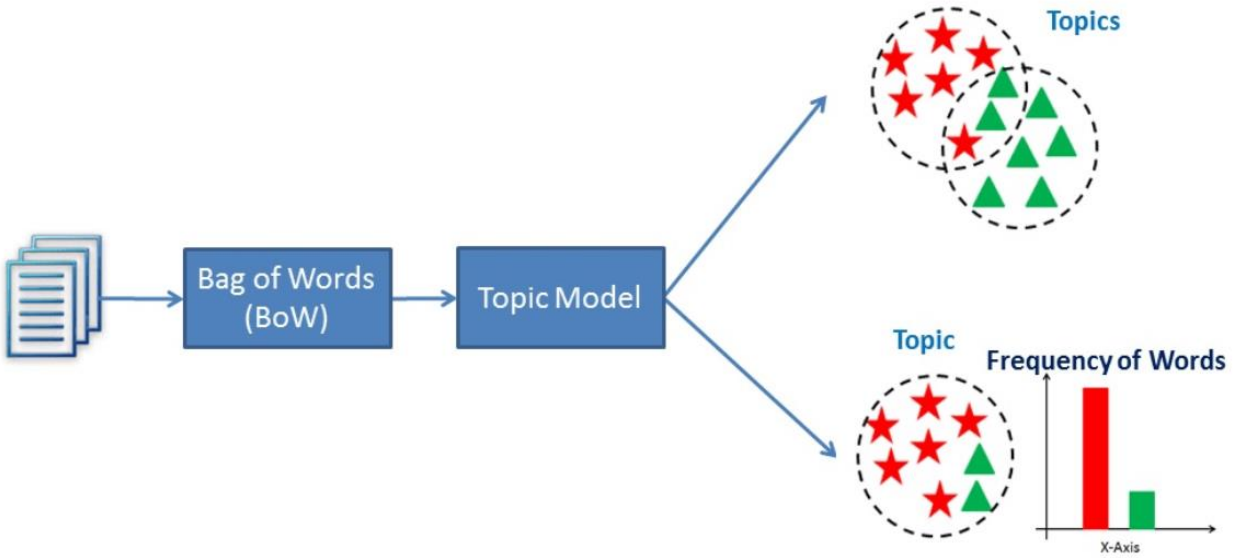
3. Latentna semantička analiza teksta

Latentna semantička analiza (engl. Latent Semantic Analysis, LSA) je tehnika obrade prirodnog jezika, za analiziranje odnosa između skupa dokumenata i termina koje oni sadrže stvarajući skup pojmova vezanih uz dokumente i pojmove (Deerwester, 2019). LSA pretpostavlja da će se riječi koje su bliske po značenju pojaviti u sličnim dijelovima teksta. Matrica koja sadrži broj riječi po paragrafu (redovi predstavljaju jedinstvene riječi i stupci predstavljaju svaki odlomak) izrađuje se iz velikog dijela teksta, a matematička tehnika koja se zove dekompozicija singularnih vrijednosti (SVD[2]) koristi se za smanjenje broja redova uz očuvanje strukture sličnosti među stupcima. Tada se uspoređuju paragrafi uzimanjem kosinusa kuta između dva nastalih bilo kojim stupcima. Vrijednosti koje su blizu 1 predstavljaju vrlo slične odlomke, dok vrijednosti blizu 0 predstavljaju vrlo različite stavke. LSA koristi model s vrećom riječi (BoW), što rezultira matricom dokumenta (pojavljivanje pojmova u dokumentu). Redovi predstavljaju pojmove, a stupci predstavljaju dokumente. LSA uči latentne teme izvođenjem matrične dekompozicije na matrici dokumenta-pojma koristeći singularnu vrijednost dekompozicije. LSA se tipično koristi kao tehnika smanjenja dimenzija ili redukcije šuma. Formula za matricu je M=UEV* gdje je M m*m matrica, U m*n lijeva singularna matrica, E je dijagonalna n*n matrica s ne negativnim realnim brojevima, V je m*n desna singularna matrica, a V* je n*m matrica (transparentna V matrica).

Otkrivanje tema (Navlani, 2018) korisno je za različite svrhe, kao što je grupiranje dokumenata, organiziranje dostupnih sadržaja na mreži za pronalaženje informacija i preporuke. Više davatelja sadržaja i novinskih agencija koriste tematske modele za preporuku članaka čitateljima. Slično tome, tvrtke za regrutiranje koriste u izvlačenju opisa poslova i preslikavanju sa skupom vještina kandidata. Ako vidite posao znanstvenika koji se bavi podacima, to je sve o izvlačenju "znanja" iz velike količine prikupljenih podataka. Obično, prikupljeni podaci su nestrukturirani. Podaci trebaju snažne alate i tehnike za analizu i razumijevanje velike količine nestrukturiranih podataka. Modeliranje tema automatski otkriva skrivene teme iz danih dokumenata. To je algoritam za analizu teksta bez nadzora koji se koristi za pronalaženje skupine riječi iz danog dokumenta. Ta

[2] SVD (Singular value decomposition) matrična metoda rastavljanja, koja predstavlja matricu u produktu dviju matrica. Nudi razne korisne primjene u obradi signala, psihologiji, sociologiji, klimi i atmosferskim znanostima, statistici i astronomiji.

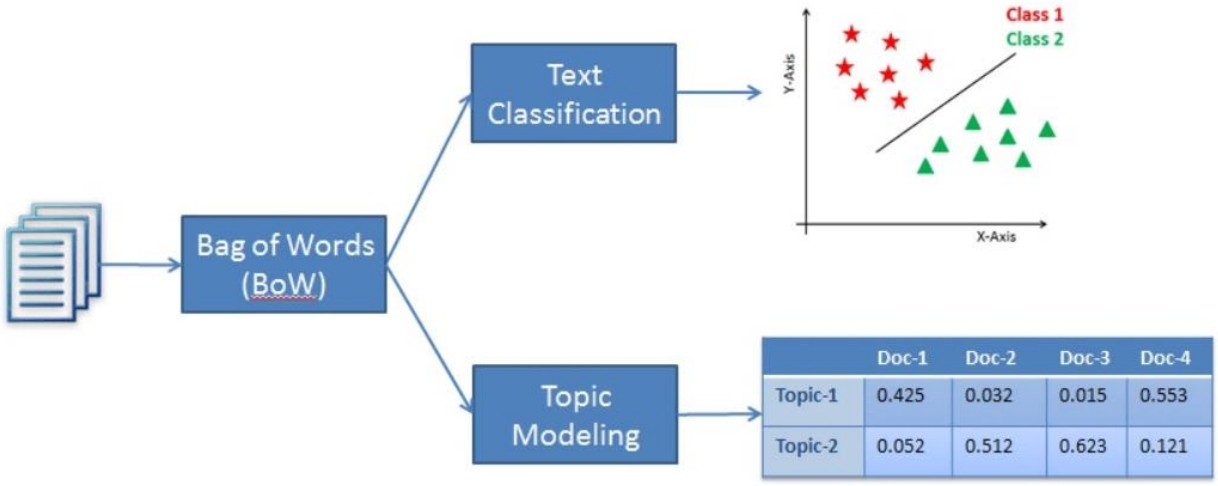
skupina riječi predstavlja temu. Postoji mogućnost da se jedan dokument može povezati s više tema. Na primjer, skupina riječi kao što su 'pacijent', 'liječnik', 'bolest', 'rak', oglas 'zdravlje' će predstavljati temu 'zdravstvo'.



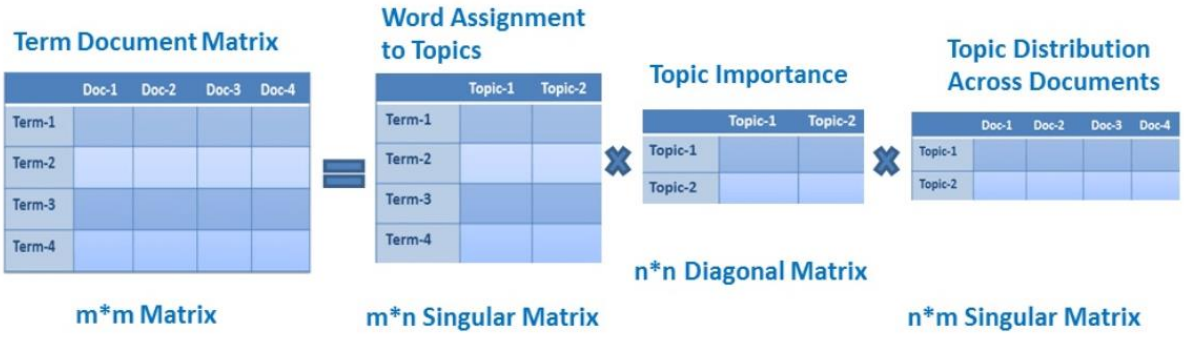
Slika 3. Set riječi (BoW) (Navlani, 2018)

Na slici 3 prikazan je skup riječi (BoW) za modeliranje tema.

Nadalje postoji način modeliranja tema klasifikacijom teksta. Klasifikacija teksta je problem strojnog učenja pod nadzorom, gdje se tekstualni dokument ili članak klasificira u unaprijed definirani skup klasa. Modeliranje tema je proces otkrivanja skupina zajedničkih riječi u tekstualnim dokumentima. Te grupne srodne riječi čine teme. To je oblik učenja bez nadzora, tako da je skup mogućih tema nepoznat. Modeliranje teme može se koristiti za rješavanje problema klasifikacije teksta. Modeliranje teme će identificirati teme koje se nalaze u dokumentu, dok klasifikacija teksta klasificira tekst u jednu klasu kao što je prikazano na slici 4.



Slika 4. Jednostavno modeliranje tema (Navlani, 2018)



Slika 5. Matrice koje se koriste u LSA (Navlani, 2018).

Na slici 5. je prikazana jednakost formule za matricu M=UEV*.

Pri utvrđivanju ili određivanju broja tema koriste se razne opcije. Jedan od načina za određivanje optimalnog broja tema je razmatranje svake teme kao klastera i utvrđivanje učinkovitosti klastera pomoću Silhouette koeficijenta. Mjera koherentnosti teme je realna mjera za utvrđivanje broja tema. Široko je korištena mjera za procjenu modela predmeta. Koristi latentne varijabilne modele. Svaka generirana tema ima popis riječi. U mjeri koherentnosti tema, naći ćete prosječnu podijeljenu s srednjom vrijednost podudarnosti riječi u parovima riječi u nekoj temi. Visoka

vrijednost modela koherencije tema smatra se dobrim tematskim modelom. LSA algoritam je najjednostavnija metoda koja je lako razumljiva i implementirana. Također nudi bolje rezultate u usporedbi s modelom vektorskog prostora. To je brže u usporedbi s drugim dostupnim algoritmima, jer uključuje samo dekompoziciju matrice. Jedini problem. LSA je dimenzija latentne teme ovisi o rangu matrice tako da ne možemo proširiti tu granicu. LSA razgrađena matrica je vrlo gusta matrica, tako da je teško indeksirati pojedinačnu dimenziju. LSA ne može uhvatiti višestruka značenja riječi.

3.1. Implementacija LSA

Pri implementaciji modela LSA na zadanom korpusu tekstova u programskom jeziku Python 3.7.1 koristi se *import* Lsi modela iz modula Gensim kako bi se dobili rezultati semantičke sličnosti riječi iz korpusa.

Nakon što su definirani i instalirani potrebni module u funkciji *load_data* dobivena je putanja i naziv datoteke korpusa koju se poziva pri pokretanju programa. U njoj je definirana lista *document_list* i lista *titles* u koje se sprema tekst. Lista dokumenata je broj dokumenata koji se koristi.

Podatci se predprocesiraju na način da se je potrebno riješiti zaustavnih riječi tj. riječi koje se filtriraju prije ili nakon obrade podataka prirodnog jezika. Potom se radi *stemming* riječi što znači da ih se reducira na njihov korijen npr. kišilo, kiši, kišiti na kiš. Procesirane riječi nakon obrade se spremaju u listu.

Funkcija *prepare_corpus* stvara matricu pojmova dokumenta i rječnik pojmova za generiranje modela. Dalje funkcija create_gensim_lsa_model nakon što smo generirali koprus može generirati LSA model korištenjem modula LsiModel iz gensima. Num_topics je broj tema gdje svaka tema ima svoj spremljeni skup riječi.

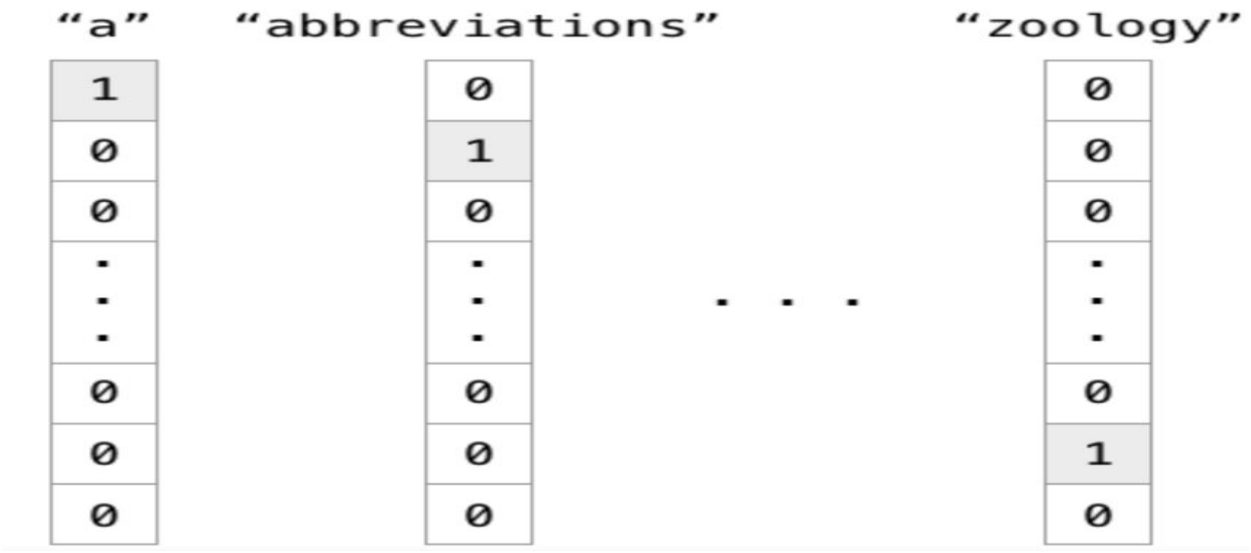
Na kraju imamo funkciju *compute_coherence_value* koja generira rezultate koherentnosti za određivanje optimalnog broja tema. Naredbom LsiModel možemo „trenirati“ model. Odabiremo broj tema, riječi, funkcijom load_data dolazimo do korpusa, čistimo ga te generiramo LSA model naredbom create_lsa_gensim_model. Pri ispisu dobivamo teme označene rednim brojem od 0 do n broja tema i vjerojatnosti sličnosti riječi u svakoj temi. (Prilog 3)

```
    """
    Input  : Broj tema i broj riječi povezanih sa svakom temom
    Purpose: izgradnja LSA modela koristeći modul gensim
    Output : LSA model
    """
    dictionary,doc_term_matrix=prepare_corpus(doc_clean)
    # generiranje LSA model
    lsamodel = LsiModel(doc_term_matrix, num_topics=number_of_topics,
id2word = dictionary)  # train model
    print(lsamodel.print_topics(num_topics=number_of_topics,
num_words=words))
    return lsamodel


def compute_coherence_values(dictionary, doc_term_matrix, doc_clean,
stop, start=2, step=3):
    """
    Ulaz   : rječnik : Gensim dictionary
             Korpus : Gensim corpus
             tekstovi: Lista ulaznih tekstova
     stop : Maksimalan broj tema
    Svrha:    Izračunati c_v coherence za različite brojeve tema
    Izlaz  :  Lista LSA tema

    """
    coherence_values = []
    model_list = []
    for num_topics in range(start, stop, step):
        # generiranje LSA model
        model = LsiModel(doc_term_matrix, num_topics=number_of_topics,
id2word = dictionary)  # train model
        model_list.append(model)
        coherencemodel = CoherenceModel(model=model, texts=doc_clean,
dictionary=dictionary, coherence='c_v')
        coherence_values.append(coherencemodel.get_coherence())
    return model_list, coherence_values

# LSA Model
number_of_topics=3
words=4
document_list,titles=load_data("","test.txt")
clean_text=preprocess_data(document_list)
```

```
model=create_gensim_lsa_model(clean_text,number_of_topics,words)
print(model)
```

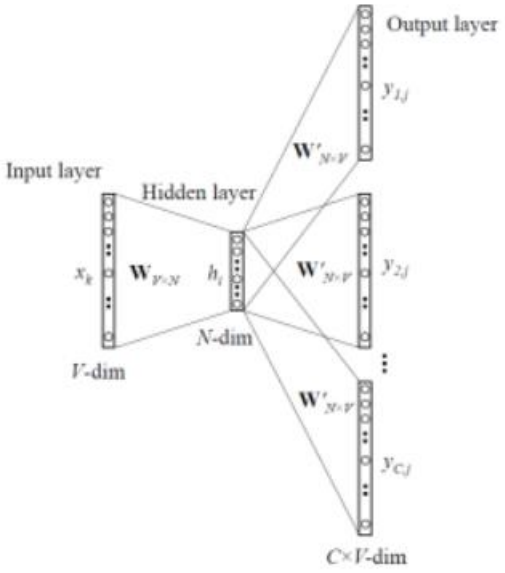
4. Word2vec

U obradi prirodnog jezika (NLP), često se preslikavaju riječi u vektore koji sadrže numeričke vrijednosti kako bi ih stroj mogao razumjeti. Ugrađivanje riječi (engl. *Word Embeddings*) je vrsta preslikavanja koja omogućuje riječima sličnog značenja sličnu reprezentaciju. Tradicionalni način predstavljanja (Slika 6.) riječi je jedan vektor, koji je vektor sa samo jednim ciljnim elementom koji je 1, a drugi je 0. Duljina vektora jednaka je veličini ukupnog jedinstvenog rječnika u korpusima. Uobičajeno, ove jedinstvene riječi kodirane su abecednim redom. Trebali bi očekivati vektore za riječi koje počinju sa slovom "a" s ciljem 1 nižeg indeksa, a one za riječi koje počinju sa slovom "z" s ciljem 1 višeg indeksa kao što je prikazano na slici 10.



Slika 6. Tradicionalan način predstavljanja riječi pomoću vekora (Huang, 2018)

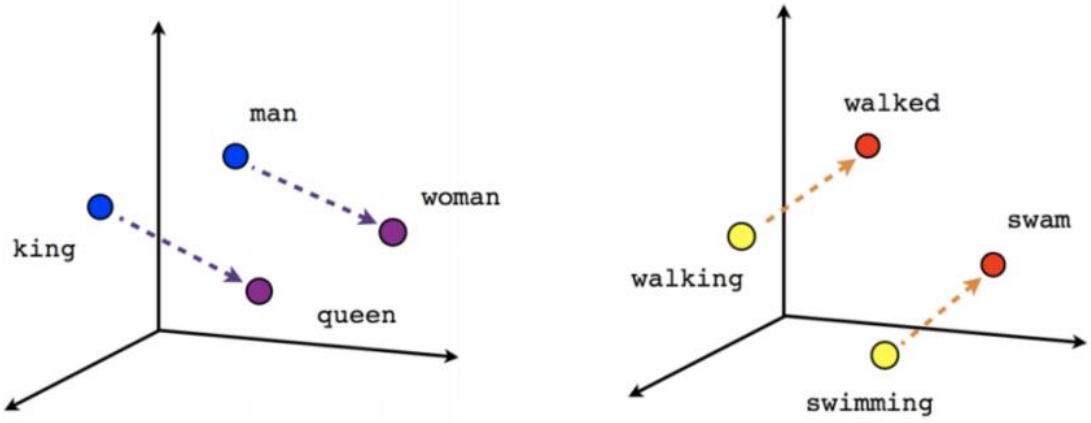
Iako je ova reprezentacija riječi jednostavna za implementaciju, postoji nekoliko problema. Ne može se zaključiti bilo koji odnos između dviju riječi s obzirom na njihovu jednoličnu reprezentaciju. Primjerice, riječi "izdržati" i "tolerirati", iako imaju slično značenje, njihove mete su daleko jedna od druge. Word2Vec (Huang, 2018) je učinkovito rješenje za te probleme koji utječu na kontekst ciljnih riječi. U osnovi, namjera je koristiti okolne riječi za predstavljanje ciljnih riječi s neuronskom mrežom čiji skriveni sloj kodira prikaz riječi. Postoje dva pristupa u realizaciji modela Word2vec: Skip-gram i kontinuirani BoW (Bag of words, CBoW). Za skip-gram, ulaz je ciljna riječ, dok su izlazi riječi koje okružuju ciljne riječi. Na primjer, u rečenici "Imam slatkog psa", ulaz bi bio "a", dok je izlaz "I", "imati", "sladak" i "pas", pod pretpostavkom da je veličina prozora pet. Svi ulazni i izlazni podaci imaju istu dimenziju. Mreža sadrži jedan skriveni sloj čija je dimenzija jednaka veličini ugradnje, koja je manja od veličine ulazno-izlaznog vektora. Na kraju izlaznog sloja primjenjuje se funkcija aktivacije, tako da svaki element izlaznog vektora opisuje koliko je vjerojatno da će se određena riječ pojaviti u kontekstu. Slika 7. prikazuje strukturu mreže.



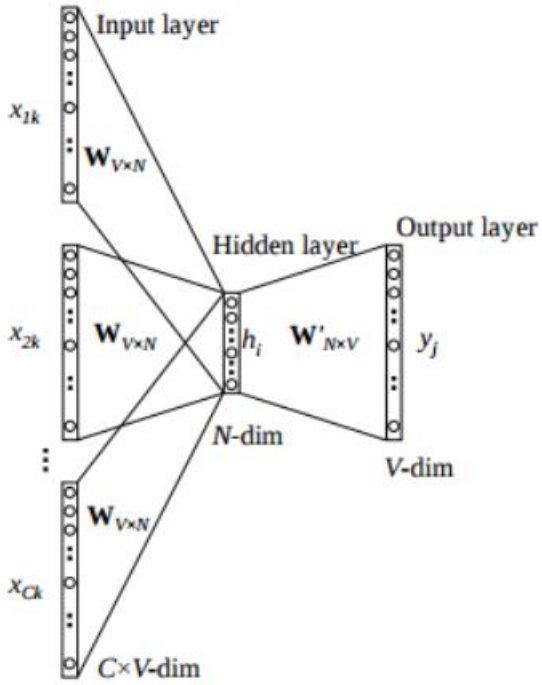
Slika 7. Struktura mreže Skip-grama (Huang, 2018)

Riječ ugrađivanja za cilj riječi može se dobiti izdvajanjem skrivenih slojeva nakon što se jedna reprezentacija te riječi unese u mrežu. S preskočenim gramom, dimenzija prikaza se smanjuje od veličine vokabulara (V) do duljine skrivenog sloja (N). Nadalje, vektori su "smisleniji" u smislu

opisivanja odnosa između riječi. Vektori dobiveni oduzimanjem dvije srodne riječi ponekad izražavaju smislen koncept kao što je spol ili glagolsko vrijeme, kao što je prikazano na slici.



Kontinuirana vreća riječi (CBOW) vrlo je slična skip-gramu no mijenja ulaz i izlaz. Ideja je da obzirom na kontekst, želimo znati koja se riječ najvjerojatnije pojavljuje u njoj.



Slika 8. Mreža CBOW (Huang,, 2018)

Najveća razlika između Skip-gram-a i CBOW-a je u tome kako se generiraju riječi vektori. Za CBOW, svi primjeri s ciljnom riječju kao ciljem ulaze u mreže i uzimaju prosjek ekstrahiranog skrivenog sloja (slika 8.). Primjerice, da bi se u rečenicama "On je dobar momak" i "Ona je mudra kraljica" izračunao prikaz riječi za riječ “a”, potrebno je navesti ova dva primjera, “On je dobar momak”, i “Ona je mudra kraljica” u neuronsku mrežu i uzeti prosjek vrijednosti u skrivenom sloju. Skip-gram unosi jednu ciljnu riječ i jedan vektor kao ulaz. Bolje radi za rijetke riječi, dok su im performanse slične.

4.1. Implementacija Word2vec

Pri implementaciji modela Word2vec na zadanom korpusu tekstova u programskom jeziku Python 3.7.1 koristi se import Word2vec modela iz modula Gensim kako bi istrenirali neuronsku mrežu i dobili rezultate semantičke sličnosti riječi iz korpusa.

U tekstualnoj datoteci test.txt nalaze se riječi za usporedbu. Najprije se ispisuje linija – potrebno je znati kakvog je tipa tekst u datoteci. Zatim se opet ulazi u datoteku i stvaraju se dvije liste. Tekst se po završetku čitanja datoteke liniju po liniju sprema i određuje se koliko će se riječi nalazi u datoteci te ih se naposljetku sprema u liste.

Čitaju se riječi u listi. Naredbom gensim.models.Word2vec izrađuje se model gdje je *documents* vokabular u kojem je spremljen korpus i isti uređen za treniranje modela, size je veličina vektora, *window* je maksimalna udaljenost trenutne i predviđene riječi, *min_count* znači da je riječi koje su se ponovile manje od dva puta potrebno izbaciti, a *workers* su niti za brže procesiranje treniranja. Potom se trenira model naredbom model.train i ispisujeu se željeni podatci za daljnju usporedbu. Naredbom model.wv.similarity pronalaze se sličnosti dviju riječi npr. riječi *man* i *woman* u korpusu.

Izračunata sličnost riječi woman i man u korpusu: 0.9427268

Naredbom wv.most_similar prikazuju se decimalne vrijednosti sličnosti zadanih riječi s drugim riječima kao što je primjer ispisa koji je stavljen u tablice gdje su u prvoj tablici ispisani top deset riječi sličnih s riječi *broher,* a tablici do nje s riječi *man*. (Prilog 1).

Top deset riječi sličnih s riječ man:	Top deset riječi sličnih s riječ brother:
('died', 0.9386678338050842),	[('applause', 0.9433990120887756),
('automobile', 0.9358378052711487),	('daughter', 0.9406968951225281),
('leon', 0.9319872260093689),	('friends', 0.9382419586181641),
('crash', 0.931269109249115),	('sopranos', 0.9327057003974915),
('brooks', 0.9271234273910522),	('husband', 0.9321234822273254),
('wife', 0.9253433346748352),	('tragedy', 0.9143981337547302),
('rohrbough', 0.9226195216178894),	('admiration', 0.9055215120315552),
('doctor', 0.9148374795913696),	('cousin', 0.9039412140846252),
('boy', 0.9143810868263245)]	('abu', 0.9035235047340393),
	('saddened', 0.9031178951263428)]

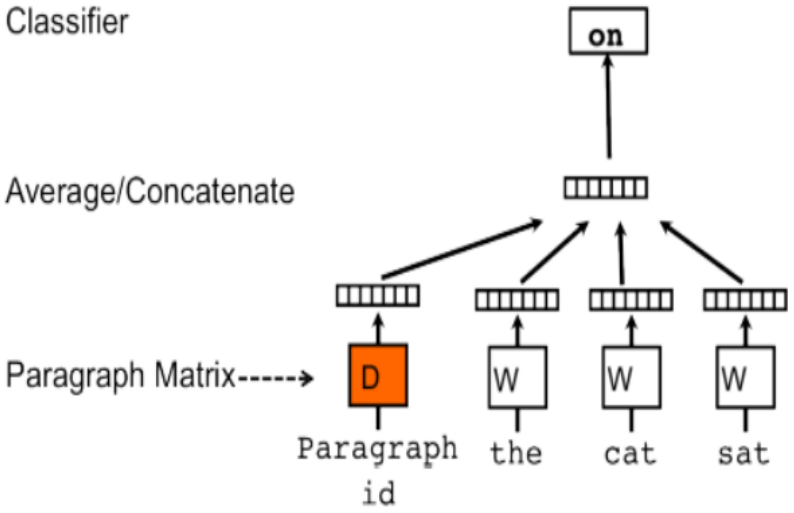
Tablica 1. Ispis najsličnijih 10 riječi s riječ man i brother

5. Doc2vec

Doc2vec je algoritam za generiranje rečeničnog vektora, stavki i dokumenata (rečenica ili paragraf). Algoritam je adaptacija modela Word2vec, koji može generirati vektore za riječi. Vektori koje generira Doc2vec mogu se koristiti za zadatke kao što je pronalaženje sličnosti između rečenica / stavaka / dokumenata. Za razliku od sekvencijskih modela kao što je RNN, gdje je niz riječi zarobljen u generiranim rečeničnim vektorima, Doc2vec rečenični vektori su redoslijedom riječi neovisni. Za zadatke sličnosti rečenica, vektori Doc2vec mogu se izvoditi

razumno dobro. Međutim, ako u ulaznom korpusu ima puno pogrešaka u pisanju, ovaj algoritam nije idealan izbor. Bolje je generirati vektore riječi izrađene od znaka n grama pomoću modela FastText (Shperber, 2017).

Cilj Doc2vec je stvoriti numerički prikaz dokumenta bez obzira na njegovu duljinu. No, za razliku od riječi, dokumenti ne dolaze u logičkim strukturama kao što su riječi tako da se mora pronaći druga metoda.



Slika 9. Doc2vec (Shperber, G., 2017)

Slika 9. prikazuje malo proširenje CBOW-a koji koristi Doc2vec. Ne koriste se riječi za predviđanje sljedeće riječi, već još jedan vektor značajki, koji je jedinstven dokument. Pri treniranju riječnih vektora W, vježba dokumenta D također je trenirana, a na kraju treninga ona sadrži numerički prikaz dokumenta. Gornji model na slici 8. naziva se distribuirana memorijska verzija vektora odlomka. Djeluje kao sjećanje koje pamti ono što nedostaje u sadašnjem kontekstu ili kao tema paragrafa. Dok vektori riječi predstavljaju pojam riječi, vektor dokumenta namjerava

predstaviti koncept dokumenta. Doc2vec model sam po sebi je metoda bez nadzora. Iz tog razloga mora se moći dodati više vektora, koji ne moraju biti jedinstveni: na primjer, ako imamo oznake za naše dokumente (kao što zapravo imamo), možemo ih dodati i dobiti njihov prikaz kao vektori.

5.1. Implementacija Doc2vec

Koriste se dvije implementacije Doc2vec-a: jedna na korištenom korpusu test.txt, a druga preko online korpusa text8.

5.1.1. Korpus text8

Podatci se spremaju u varijablu dataset u koju je naredbom api.load skinuta i učitaan korpus text8. U funkciji create_tagged_document kreira se lista označenih dokumenata (rečenice ili paragrafi) i pripremaju se podatci za treniranje. Naredbom gensim.model.Doc2vec. Doc2vec inicijalizira se Doc2vec model gdje je vector_size broj vektora u koji se spremaju rečenice, min_count ignorira riječi koje su se ponovile manje od n puta, u ovom slučaju dva, a epochs je cjelobrojna vrijednost iteracija korpusa. Naredbom model.build_vocab kreira se vokabular od podataka za treniranje i naredbom model.train trenira se model liste označenih dokumenata (train_data). Naredbom model.infer_vector iz korpusa se dobivaju rezultate rečeničnog vektora zadanih riječi u rečenici te s naredbom model.dovecs .most_similar([vector]) tagove i sličnosti najsličnijih riječi (Prilog 2).

Vjerojatnosti sličnosti:
[(1678, 0.9957818984985352)
(1680, 0.9903787970542908)
(1675, 0.9346541166305542)
(1690, 0.9268406629562378)
(1568, 0.852929413318634),
(1412, 0.8474373817443848)
(1556, 0.836499035358429)
(1558, 0.8357431888580322),
(381, 0.8249835968017578)
(1522, 0.7982527017593384)]

Tablica 2. Vjerojatnosti najsličnijih rečenica u korpusu text8

5.1.2. Korpus test2.txt

U varijabli *documents* kreira se lista označenih dokumenata (rečenice ili paragrafi) i pripremaju se podatci za treniranje. Model se trenira naredbom Doc2vec gdje je *documents* lista označenih dokumenata (rečenica), *vektor_size* veličina vektora u kojem spremamo rečenice, *min_count* izbacuje riječi koje su se ponovile manje od jednog puta, a *workers* su niti za brže treniranje modela. Podatci se učitani iz korpusa test2.txt s naredbom za čitanje datoteke. Naredbom model.infer_vector iz korpusa dobivaju se rezultati rečeničnog vektora zadanih riječi, a naredbom model.docvecs.most_similar dobivaju se vjerojatnosti sličnosti u rečenicama iz korpusa s time da možemo odabrati top 5 rečenica (Prilog 2).

Vjerojatnosti top 5 najsličnijih rečenica:
[(7, 0.9835432767868042)
(4, 0.9048029184341431)
(3, 0.6851617097854614)
(1, 0.6681811213493347)
(5, 0.5188946723937988)]

Tablica 3. Vjerojatnosti sličnosti najsličnijih rečenica u korpusu test2.txt

6. GloVe

Drugi dobro poznati model koji uči vektore ili riječi iz informacija o zajedničkom pojavljivanju, tj. koliko često se pojavljuju zajedno u velikim tekstualnim korpusima je Globalni Vektor (GloVe). Dok je Word2vec neuronska mreža koja uči vektore za poboljšanje predviđajuće sposobnosti, GloVe (Sciforce, 2018) je model koji se temelji na brojanju. Općenito govoreći, modeli zasnovani na brojanju uče vektore radeći smanjenje dimenzije na matrici brojeva pojavljivanja. Prvo konstruiraju veliku matricu informacija o zajedničkom pojavljivanju, koja sadrži informacije o tome koliko često se svaka riječ pohranjena u redovima vidi u nekom kontekstu tj. stupcima. GloVe je u osnovi log-bilinearni model s ciljem dobivanja najmanjih kvadrata. Model se zasniva na jednostavnoj ideji usporedbe vjerojatnosti koincidencije riječi – riječi imaju potencijal za kodiranje nekog oblika značenja koje se može kodirati kao vektorske razlike. Slijedi da je cilj obuke naučiti vektore riječi tako da njihov točkasti proizvod bude jednak logaritmu vjerojatnosti koincidencije riječi. Budući da je logaritam omjera jednak razlici logaritama, ovaj cilj povezuje omjere vjerojatnosti koincidencije s vektorskim razlikama u prostoru riječnog vektora. Stvaraju se vektori riječi koji se dobro izvode na zadatcima analogije riječi i na zadatcima sličnosti i prepoznavanju imenovanih entiteta. Gotovo sve metode bez nadzora za učenje riječnih prikaza koriste statistiku pojmova riječi u korpusu kao primarni izvor informacija, ali se javlja problem kako generirati smisao iz tih statistika i kako bi riječni vektori mogli predstavljati to značenje. Uzmimo za primjer riječi led i para. Odnos ovih riječi može se otkriti proučavanjem omjera vjerojatnosti njihove pojave s različitim sondama, k. Neka je P (k | w) vjerojatnost da se riječ k pojavljuje u kontekstu riječi w: led se češće pojavljuje s krutim nego s plinom, dok se para češće pojavljuje s plinom nego s krutim, obje se riječi često pojavljuju s i rijetko s nepovezanom modom riječi. P (čvrsti | led) će biti relativno visok, a P (čvrsta | para) će biti relativno niska. Iz navedenog proizlazi da će omjer P (krutog | leda) / P (krutog | para) biti velik. Ako uzmemo riječ kao što je plin koji je vezan za paru, a ne za led, odnos P (plin | led) / P (plin | para) će umjesto toga biti mali. Za riječ koja se odnosi na led i paru, kao što je voda, očekujemo da će omjer biti blizu jedan.

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

Slika 10. Formula za minimiziranje objektivne funkcije J (Sciforce, 2018)

Način na koji GloVe predviđa okolne riječi je maksimiziranje vjerojatnosti pojavljivanja kontekstualne riječi s obzirom na središnju riječ izvedbom dinamičke logističke regresije. Prije treniranja stvarnog modela konstruira se matrica pojavljivanja X, gdje je stanica Xij snaga koja predstavlja koliko često se riječ i pojavljuje u kontekstu riječi j. Nakon što je X spreman, potrebno je odrediti vektorske vrijednosti u kontinuiranom prostoru za svaku riječ u korpusu, drugim riječima, za izgradnju vektora riječi koji pokazuju kako se svaki par riječi i, j pojavljuju zajedno. Proizvode se vektori s mekim ograničenjem za svaki par riječi i te riječi j gdje su bi i bj su skalarni izrazi pristranosti koji su povezani s riječima i te j. Radi se minimizacija objektivne funkcije J, koja vrednuje zbroj svih kvadratnih pogrešaka, dane funkcijom f: gdje je V veličina vokabulara. Model (slika 10.) generira dva skupa riječi, Wi i Wj. Kada je X simetrična, Wi i Wj su ekvivalentni i razlikuju se samo kao rezultat njihove slučajne inicijalizacije. Dva skupa vektora rade jednako. Model GloVe koristi glavnu korist od brojevnih podataka. Sposobnost hvatanja globalne statistike znači da istovremeno hvata značajne linearne pod-strukture koje prevladavaju u log-bilinearnim metodama temeljenim na predviđanju kao što je Word2vec. Kao rezultat toga, GloVe je globalni log-bilinearni regresijski model za nenadzirano učenje reprezentacija riječi koji nadmašuje druge modele na analogiji riječi, sličnosti riječi i zadacima prepoznavanja imenovanih entiteta. Prednosti GloVe modela su da je treniranje podataka brzo, skalabilan je na velikom skupu, ima dobre performanse na korpusima te malim vektorima riječi i može se rano zaustaviti treniranje podataka. Mana modela je korištenje mnogo memorije.

7. Usporedba LSA vs Deep Learning models

Modeli dubokog učenja (Word2vec i Doc2vec) su modeli koji se temelje na predviđanju. Za dani vektor riječi (Word2vec) ili rečenica (Doc2vec) može se predvidjeti kontekst riječi ili rečenica vektora. LSA model se temelji na brojanju gdje slični izrazi imaju iste vrijednosti za različite dokumente. Modeli temeljeni na dubokom učenju dali su bolje rezultate. Kao primjer uzet je test2.txt korpus s četrnaest testnih rečenica i uspoređeni su rezultati Doc2vec i LSA. Rezultati se dobivaju pokretanjem kodova zapisanih u popisu priloga (LSA i kod Doc2vec vezan za korpus text2), a rečenice iz korpusa i vjerojatnosti sličnosti modela su postavljene u tablicu 4. U LSA skup riječi podijeljen je u teme koje su bile vezane za rečenice te ih je trebalo prepoznati u ispisu programskog koda. U njima je sadržan skup riječi koje sadrže vjerojatnosti riječi koja pripada određenoj temi. U Doc2vec modelu, koji se temelji na treniranoj neuronskoj mreži, rečenice su bile podijeljene u tagove te se svaki tag odnosio na određene rečenice. Kosinusna sličnost doc2vecs.most_similar računa sličnost u Doc2vec modelu. Decimalne vjerojatnosti predstavljaju koliko su rečenice iz zadanog taga Doc2veca slične. Ako je kosinusna sličnost veća od 0.5, rečenice iz tablice 4. su pogođene kao slične.

7.1. Rezultati

Rečenice	*Doc2vec*	*LSA*
Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence. Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.	0.387	0.319
Yucaipa *owned* Dominick's before selling the chain to Safeway in 1998 for $2.5 billion. Yucaipa bought Dominick's in 1995 for $693 million and sold it to Safeway for $1.8 billion in 19	0.439	0.221-0-234

They had published an advertisement on the Internet on June 10, offering the cargo for sale, he ad On June 10, the ship's owners had published an advertisement on the Internet, offering the explosives for sale.	0.998	0.179-0.272
Around 0335 GMT, Tab shares were up 19 cents, or 4.4%, at A$4.56, having earlier set a record high of A$4.57. Tab shares jumped 20 cents, or 4.6%, to set a record closing high at A$4.57.	0.881	0.169-0.508
The stock rose $2.11, or about 11 percent, to close Friday at $21.51 on the New York Stock Exchange. PG&E Corp. shares jumped $1.63 or 8 percent to $21.03 on the New York Stock Exchange on Friday.	0.991	0.156-0.464
Revenue in the first quarter of the year dropped 15 percent from the same period a year earlier. With the scandal hanging over Stewart's company, revenue the first quarter of the year dropped 15 percent from the same period a year earlier.	0.692	0.156-0.466
The Nasdaq had a weekly gain of 17.27, or 1.2 percent, closing at 1,520.15 on Friday. The tech-laced Nasdaq Composite .IXIC rallied 30.46 points, or 2.04 percent, to 1,520.15.	0.656	0.221-0.234

Tablica 4. Rezultati sličnosti rečenica iz korpusa test2.txt u modelu Doc2vec i težina riječi u rečenicama u LSA

8. Zaključak

Mjerenje semantičke sličnosti tekstova primjenom modela dubokog učenja je zanimljiva tema koja ulazi u različita područja računalne analize prirodnog jezika. Semantičke mjere odličan su alat za obradu jezičnih jedinica i njihovih instanci. Koristeći ih, otvaraju nam se razne mogućnosti koje nude jezične jedinice. One opisuju semantičke sličnosti, nude razne obrade na raznim tekstovima ili korpusima te mogu izraditi razne semantičke modele pomoću znanja, raznih ontologija i usporedbi jezičnih jedinica.

Obrađeni modeli prikazuju vjerojatnosti koje utvrđuju sličnosti tekstova. Cilj je bio istražiti modele koji se temelje na dubokom učenju (Doc2vec, Word2vec, GloVe, Fasttext i dr.) i usporediti ih s modelom koji se temelje na latentnoj semantičkoj analizi (LSA).

Za izradu modela bilo je potrebno pronaći odgovarajuće modele na internetu, izraditi ih u programskom jeziku Python ili osposobiti već gotov kod s neke internetske stranice na popularnom korpusu tekstova za testiranje semantičke sličnosti (test.txt) ili korpusu skinutom s interneta (text8). Svi izrađeni modeli korišteni su pomoću biblioteke gensim koja sadrži module za navedene modele.

LSA model predstavlja kontekstualno značenje riječi statističkim putem računanja izvršenih na korpusu dokumenata. Cilj je dobiti sveukupnost informacija o svim riječima kontekstima u kojima se određena riječ pojavljuje ili ne pojavljuje te pruža skup međusobnih ograničenja koja određuju sličnosti značenje riječi.

Word2vec model je neuronska mreža s dvije razine, koja implementira neprekidne vreće riječi (BoW) i skip-gram arhitekture za računanje vektorskih prikaza riječi ili njihovog konteksta i kao izlaz daje vokabular riječi iz korpusa s njihovim vektorskim prikazima.

Doc2vec model mijenja Word2Vec model u nenadzirano učenje kontinuiranih prikaza (reprezentacija) za veće blokove teksta, kao što su rečenice, paragrafi ili cijeli dokumenti, a GloVe i Fasttext su ekstenzije modela Word2vec napravljene za lakše i brže treniranje većeg skupa podataka.

Pomoću opisanih i implementiranih modela istraženi su rezultati mjerenja sličnosti riječi i rečenica u korpusu. Prikazane tablice su napravljene iz programskih kodova (u prilozima). Uspoređeni su modeli koji se temelje na dubokom učenju s modelom LSA. Modeli s treniranom neuronskom mrežom (Word2vec, Doc2vec) se izrađuju uz manje programskog koda, rade bolje na većim skupovima podataka i nude više različitih opcija za lakše, brže i bolje demonstriranje na zadacima poput izražavanja analogija, mjerenja sličnosti, uspoređivanja pojmova i dr.

9. Literatura

Atish Pawar, V. M. (n.d.). https://arxiv.org/pdf/1802.05667.pdf

Deerwester, S. D. (2019). http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf

Dict, O. (2012). https://www.amazon.com/New-Oxford-Dictionary-Writers-Editors/dp/0198610408

Hahn, C. R. (2003). https://www.dectech.co.uk/publications/LinksNick/CategorizationPerceptionAndMemory/Similarity%20as%20transformation.pdf

Huang, S. (February 2018). *Word2vec* : https://towardsdatascience.com/word-embedding-with-word2vec-and-fasttext-a209c1d3e12c?fbclid=IwAR3koRGA7mL6SgAEMlg8Z4t8MTUIQIxuLEWZjajfl38F4ArrEoEXtKOckaw

Library, G. (n.d.). *Gensim*: https://radimrehurek.com/gensim/intro.html

Markman, A. b. (1993). http://groups.psych.northwestern.edu/gentner/papers/MarkmanGentner93c.pdf

McCarthy, J. M. (2006). file:///C:/Users/Ernest/Downloads/1904-Article%20Text-1900-1-10-20080129.pdf

Michael Campr, K. J. (n.d.). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.722.6114&rep=rep1&type=pdf

Navlani, A. (2018, October). *LSA*. https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python?fbclid=IwAR0iwS_pjBewvkOho6pN4Y-so-n4-XLuLOd1aoL2uPazuFHfTdXXQU9KT3U

Ramzan, A. (2016). https://pdfs.semanticscholar.org/d91b/1b996678425418321d9de5b251778eb506d7.pdf

Rissland, E. (2006). Umjetna inteligencija i sličnosti: https://www.researchgate.net/publication/3454358_AI_and_Similarity

S´ebastien Harispe, S. R. (2017). file:///C:/Users/Ernest/Downloads/harispe_Semantic%20similarity%20from%20natural%20language%20and%20ontology%20analysis_2017%20(1).pdf

Sciforce. (2018). https://medium.com/sciforce/word-vectors-in-natural-language-processing-global-vectors-glove-51339db89639

Shepard. (1987). http://cseweb.ucsd.edu/~gary/PAPER-SUGGESTIONS/shepard-science-87.pdf

Shperber, G. (July 2017). *Doc2vec:* https://medium.com/scaleabout/a-gentle-introduction-to-doc2vec-db3e8c0cce5e

Tversky. (2004). http://cseweb.ucsd.edu/~gary/PAPER-SUGGESTIONS/Preference,%20Belief,%20and%20Similarity%20Selected%20Writings%20%28Bradford%20Books%29.pdf

10. Popis priloga

Prilog 1: Kod Word2vec

```
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 21 23:16:29 2019

@author: Ernest
"""
import gensim
import logging
from gensim.models.word2vec import Word2Vec


logging.basicConfig (format="%(asctime)s : %(levelname)s : %(message)s", level=logging.INFO)
data_file="test.txt"
with open ("test.txt", 'r',encoding='utf-8') as f:
        for i,line in enumerate (f):
            print(line)
            break
def read_input(input_file):


    logging.info("učitavanje datoteke {0}...ovo može potrajati".format(input_file))
    with open(input_file, 'r',encoding='utf-8') as f:
        documents_list = []
        titles=[]
        for line in f.readlines():
            text = line.strip()
            documents_list.append(text)
            yield gensim.utils.simple_preprocess(line)
        print("Ukupan broj riječi:", len(documents_list))
        titles.append(text[0:min(len(text), 100)])
        return documents_list, titles

documents = list (read_input(data_file))
logging.info ("Gotovo čitanje datoteke")
model = gensim.models.Word2Vec(documents, size=150, window=10, min_count=2, workers=10)
model.train(documents,total_examples=len(documents),epochs=10)
a=["brother"]
b=["man"]
c=model.wv.most_similar (positive=a, topn=10)
d=model.wv.most_similar(b, topn=10)
```

```
z=model.wv.similarity('woman', 'man')

print("Riječi slične s riječ brother i njihove vjerojatnosti: ",c)
print("Riječi slične s riječ man i njihove vjerojatnosti: ",d)
print("Izračunata sličnost riječi woman i man u korpusu: ",z)
```

Prilog 2: Kodovi modela Doc2vec

Korpus: Text8:

```
import gensim
import gensim.downloader as api

# Skidanje skupa podataka (korpus text8)
dataset = api.load("text8")
data = [d for d in dataset]
# Izradnja tagiranih rečenica za Doc2Vec
def create_tagged_document(list_of_list_of_words):
    for i, list_of_words in enumerate(list_of_list_of_words):
        yield gensim.models.doc2vec.TaggedDocument(list_of_words, [i])

train_data = list(create_tagged_document(data))

print(train_data[:1])
# Inicijalizacija Doc2Vec model
model = gensim.models.doc2vec.Doc2Vec(vector_size=3, min_count=0, epochs=1)

# Izradnja vokabulara (riječnika)
model.build_vocab(train_data)

# Treniranje Doc2Vec model
model.train(train_data, total_examples=model.corpus_count, epochs=model.epochs)
vector=model.infer_vector(['a','but'])
sims = model.docvecs.most_similar([vector])
print ("Vjerojatnosti sličnosti od rečenica: ",sims)
```

Korpus: Test2.txt

```
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
from collections import namedtuple
fname=open("test2.txt","r")
docs = []
documents = namedtuple('AnalyzedDocument', 'words tags')
for i, text in enumerate(fname):
    words = text.lower().split()
    tags = [i]
    docs.append(documents(words, tags))

print(docs)
model = Doc2Vec(docs, vector_size=2, window=2, min_count=1, workers=4)
print(model)

vector = model.infer_vector(fname)
print(vector)
```

```
simils = model.docvecs.most_similar([vector])
sims = model.docvecs.most_similar([vector],topn=5)
print("Vjerojatnosti sličnosti od rečenica: ", simils)
print("Vjerojatnosti slicnosti top 5 rečenica: ", sims)
```

Prilog 3: Kod LSA

```
import os.path
from gensim import corpora
from gensim.models import LsiModel
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from gensim.models.coherencemodel import CoherenceModel
import matplotlib.pyplot as plt
import pyLDAvis.gensim


def load_data(path,file_name):
    documents_list = []
    titles=[]
    print(os.path.join(path,file_name))
    f=open(file_name,"r",encoding="utf-8")
    for line in f.readlines():
        text = line.strip()
        documents_list.append(text)
        documents_list.append(text)
    print("Ukupan broj rečenica u datoteci:",len(documents_list))
    titles.append( text[0:min(len(text),100)] )
    return documents_list,titles



def preprocess_data(doc_set):
    """
    Input  : lista dokumenata (rečenice, paragrafi)
    Svrha  :  Predprocesiranje (tokeniziranje, rješavanje stopwordsa i stemming)
    Output : predprocesiran tekst
    """
    # inicijalizacija regexa
    tokenizer = RegexpTokenizer(r'\w+')
    # Izgradnja engleskih stopwordsa
    en_stop = set(stopwords.words('english'))
    # kreiranje stemmera
    p_stemmer = PorterStemmer()
    # lista tokeniziranih riječi u setu dokumenata
    texts = []
    for i in doc_set:
        # očisti i tokeniziraj string
        raw = i.lower()
        tokens = tokenizer.tokenize(raw)
        # ukloni stopwordse iz riječi
        stopped_tokens = [i for i in tokens if not i in en_stop]
        # Stemmiranje riječi
        stemmed_tokens = [p_stemmer.stem(i) for i in stopped_tokens]
        # dodaj riječ u praznu listu text
        texts.append(stemmed_tokens)
    return texts
```

```
def prepare_corpus(doc_clean):
    """
    svrha: Izgradnja rječnika
    izlaz : Rječnik i matrica pojmova (riječi)
    """
    # Izgradnja rječnika di je svakoj riječi dodan ID,  dictionary = corpora.Dictionary(doc_clean)
    dictionary = corpora.Dictionary(doc_clean)
    # Konvertiranje liste dokumenata(korpus) u matricu dokumenata koristeći napravljeni riječnik.
    doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
    # generiranje LDA model
    return dictionary,doc_term_matrix

def create_gensim_lsa_model(doc_clean,number_of_topics,words):
    """
    Input  : Broj tema i broj riječi povezanih sa svakom temom
    Purpose: izgradnja LSA modela koristeći modul gensim
    Output : LSA model
    """
    dictionary,doc_term_matrix=prepare_corpus(doc_clean)
    # generiranje LSA model
    lsamodel = LsiModel(doc_term_matrix, num_topics=number_of_topics, id2word = dictionary)  # train model
    print(lsamodel.print_topics(num_topics=number_of_topics, num_words=words))
    return lsamodel


def compute_coherence_values(dictionary, doc_term_matrix, doc_clean, stop, start=2, step=3):
    """
    Ulaz   : rječnik : Gensim dictionary
             Korpus : Gensim corpus
             tekstovi: Lista ulaznih tekstova
     stop : Maksimalan broj tema
    Svrha:    Izračunati c_v coherence za različite brojeve tema
    Izlaz  :  Lista LSA tema

    """
    coherence_values = []
    model_list = []
    for num_topics in range(start, stop, step):
        # generiranje LSA model
        model = LsiModel(doc_term_matrix, num_topics=number_of_topics, id2word = dictionary)  # train model
        model_list.append(model)
        coherencemodel = CoherenceModel(model=model, texts=doc_clean, dictionary=dictionary, coherence='c_v')
        coherence_values.append(coherencemodel.get_coherence())
    return model_list, coherence_values


# LSA Model
number_of_topics=10
words=7
document_list,titles=load_data("","test2.txt")
clean_text=preprocess_data(document_list)
model=create_gensim_lsa_model(clean_text,number_of_topics,words)
print(model)
```

11. Popis slika:

12. Popis tablica: